

AMENDMENTS TO THE CLAIMS:

Please amend claim 1 as follows. This listing of claims will replace all prior versions and listings of claims in the application:

1. (Currently Amended) A computer-readable medium, operative to serve as a database interface, having instructions which when executed by a computer system, comprise the following steps:

receiving a Structured Query ~~Logic~~ Language (SQL) call at the computer system;
mapping the SQL call to a general computer language programming call of a computer application; and

executing the general computer language programming call to invoke functions of the computer application that correspond to functions specified by the SQL call.

2. (Original) The medium of claim 1 wherein said general computer language programming call is an Enterprise Java Bean (EJB) call.

3. (Cancelled).

4. (Original) The medium of claim 1 wherein the computer system is an application server.

5. (Previously Presented) The medium of claim 4 wherein the application server receives the SQL call from a client computer system.

6. (Original) The medium of claim 1 further comprising generating a database call to a database in response to executing the general computer language programming call.

7. (Cancelled).

8. (Previously Presented) The medium of claim 6 further comprising validating data operation prior to issuing an SQL call to a database.

9. (Original) The medium of claim 1 wherein the general computer programming language has components that generate database calls to a database in response to executing the general computer language programming calls; and further comprising:

analyzing the components to determine the correspondence between the database elements and the elements of the components that access the database elements; and

creating a database bridge map that identifies the correspondence.

10. (Previously Presented) The medium of claim 9 wherein the step of mapping utilizes the database bridge map to map the SQL call to a general programming language call.

11. (Original) The medium of claim 10 wherein the computer programming language is object-oriented and wherein said components are objects.

12. (Previously Presented) The medium of claim 11 wherein the step of analyzing the components comprises determining the methods that are invoked in the objects for use in determining said correspondence.

13. (Original) The medium of claim 12 wherein the methods are identified by searching for a method of the form <command prefix>XXX.

14. (Original) The medium of claim 13 wherein the <command prefix> is "get".

15. (Original) The medium of claim 13 wherein the <command prefix> is "set".

16. (Previously Presented) The medium of claim 12 wherein design patterns are used to map the received SQL call to the general programming call.

17. (Original) The medium of claim 16 wherein said general computer programming language is the Java programming language.

18. (Previously Presented) The medium of claim 1, wherein the SQL call received at the computer system is a first SQL database call and a column layout specified in the first SQL database call is different than a second SQL database call generated to a SQL database in response to executing the general computer language programming call.

19. (Original) A computer implemented method for accessing a database, comprising the steps of:

receiving a database protocol command for accessing elements in a database;

converting said database protocol command into a general computer programming language command for accessing said database; and

accessing said database by executing said general computer programming language command.

20. (Original) The method of claim 19 wherein said database protocol command is received from a computer application executing on a first computer system.

21. (Original) The method of claim 20 wherein said computer application is in a different computer programming language than said general computer programming language.

22. (Original) The method of claim 21 wherein said step of converting comprises mapping said database protocol command to a general computer programming language command.

23. (Previously Presented) The method of claim 19 wherein said database protocol command is an SQL call.

24. (Original) The method of claim 23 wherein the general computer programming language is the JAVA programming language.

25. (Original) The method of claim 24 wherein said computer application is a Visual Basic computer application.

26. (Previously Presented) A computer readable medium, for accessing a database, comprising instructions which when executed by a computer, comprise the following steps:

 exposing software components, in a first computer programming language, of an application server as database elements, said software components being operative for accessing said database;

 receiving a database protocol command for accessing the database at the application server;

 converting the database protocol command to a command syntax of the first computer programming language corresponding to at least a selected one of said software components; and

 accessing said database using said selected one of said software components.

27. (Original) The medium of claim 26 wherein said database protocol command is an SQL command.

28. (Original) The medium of claim 26 wherein the first computer programming language is the Java programming language.

29. (Original) The medium of claim 26 wherein the first computer programming language has components that generate database calls to a database in response to executing said selected one; and further comprising:

analyzing the components to determine the correspondence between the database elements and the elements of the components that access the database elements;

creating a database bridge map that identifies the correspondence.

30. (Original) The medium of claim 29 wherein the computer programming language is object-oriented and wherein said components are objects.

31. (Original) The medium of claim 30 wherein the step of analyzing the components comprises determining the methods that are invoked in the objects for use in determining said correspondence.

32 (Previously Presented) A system for interfacing between a computer and a database, comprising:

a command converter operative to convert a first database programming language call received from the computer to a general computer programming language call that corresponds to the database programming language call; and wherein

said system is operative to execute said general computer programming language call and operative to generate a second database programming language call, which corresponds to the first database programming language call, to access a database.

33. (Original) The system of claim 32 wherein said system is an application server.

34. (Previously Presented) The system of claim 33 wherein said computer is a client computer that generates said first database programming language call.

35. (Original) The system of claim 34 wherein said database is accessed through a database server.

36. (Previously Presented) The system of claim 32 wherein said first and second database programming language calls are SQL calls.

37. (Previously Presented) The system of claim 36 wherein said first and second database programming language calls specify different column names.

38. (Original) The system of claim 36 wherein said general programming language is an object-oriented computer programming language.

39. (Original) The system of claim 37 wherein said programming language is the Java programming language.

40. (Previously Presented) The system of claim 32 wherein said command converter comprises a mapping module that maps said first database programming language call to said general computer programming language call.